

What
lea
exe

[illegible]

5 each of said at least one terminal apparatus
comprises a message transmitting unit which transmits a
message containing version information indicating a
program version; and

each of said at least one program execution

10 apparatus comprises,

a message receiving unit which receives said message containing version information indicating a program version, from one of said at least one terminal apparatus,

15 a program storing unit which stores one
or more program components,

a pre-transfer information management table which holds information on said one or more program components stored in said program storing unit,

20 a program memory unit which is allocated
to an activated process, and temporarily stores at least
one program component transferred from said program
storing unit,

a post-transfer information management
25 table which holds information on said at least one program
component stored in said program memory unit, and

a program executing unit \ which

July 1972

dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to
5 enable execution of said one of said one or more program components in said process.

2. An information processing system according to claim 1, wherein said post-transfer information management
10 table indicates for each of said at least one program component a reference count which indicates the number of executions in which said each of said at least one program component are currently referred to.

3. An information processing system according to claim 2, wherein said program executing unit removes one of said at least one program component from said program memory unit when said reference count for said one of said
15 at least one program component is zero, and information on a newer version of said one of said at least one program component is included in the said post-transfer information management table.
20

4. An information processing system according to claim 1, wherein said message received by said message receiving unit further contains an evaluation flag which indicates whether or not a program component is under
25

last
evaluation.

5. An information processing system according to claim 4, wherein said program executing unit executes said program component for which said evaluation flag is contained in said message, in a separate process, when the evaluation flag indicates that said program component for which said evaluation flag is contained in said message is under evaluation.

6. An information processing system according to claim 1, wherein said post-transfer information management table indicates for each of said at least one program component an evaluation count which indicates the number of evaluations which are currently performed on said each of said at least one program component.

7. An information processing system according to claim 6, wherein said program executing unit terminates execution of one of said at least one program component in a separate process when said evaluation count corresponding to said one of said at least one program component is zero.

8. An information processing system according to claim 1, further comprising a management apparatus which sets and manages said one or more program components

Sub a2
stored in said program storing unit and said information on said one or more program components held in said pre-transfer information management table.

5 9. A terminal apparatus comprising:

an interface unit which controls operations of inputting and outputting software; and

a message transmitting unit which transmits a message containing version information
10 indicating a program version.

Sub a2
15 10. A program execution apparatus comprising:

a message receiving unit which receives a message containing version information indicating a
15 program version;

a program storing unit which stores one or more program components;

a pre-transfer information management table which holds information on said one or more program
20 components stored in said program storing unit;

a program memory unit which is allocated to an activated process, and temporarily stores at least one program component transferred from said program storing unit;

25 a post-transfer information management table which holds information on said at least one program component stored in said program memory unit; and

Sub

a program executing unit which dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to enable execution of said one of said one or more program components.

11. A distributed processing system including at least one client apparatus and a plurality of server apparatuses, and realizing an N-tier client-server environment, wherein

each of said at least one client apparatus comprises a client stub processing unit which executes stub processing of a first message which contains version information indicating a program version; and

each of said plurality of server apparatuses comprises,

a server skeleton processing unit which executes skeleton processing of said first message containing version information indicating a program version,

a distributed-object storing repository which stores one or more distributed objects,

a pre-transfer information management table which holds information on said one or more distributed objects stored in said distributed-object

Sub
a2

storing repository,

a memory which is allocated to an activated process, and temporarily stores at least one distributed object transferred from said distributed-
5 object storing repository,

a post-transfer information management table which holds information on said at least one distributed object stored in said memory,

00720045-120400
a distributed object execution control
10 unit which dynamically links one of said one or more distributed objects corresponding to said version information contained in said first message of which skeleton processing is executed by said server skeleton processing unit, to said memory, so as to enable execution
15 of at least one function in said one of said one or more distributed objects, and

a server stub processing unit which executes stub processing of a second message containing said version information so as to transmit the second
20 message to another of said plurality of server apparatuses.

12. A distributed processing system according to claim 11, wherein said post-transfer information management table indicates for each of said at least one
25 distributed object a reference count which indicates the number of executions in which said each of said at least one distributed object are currently referred to.

July 1982

13. A distributed processing system according to claim 12, wherein said distributed object execution control unit removes one of said at least one distributed object from said memory when said reference count for said one of said at least one distributed object is zero, and information on a newer version of said one of said at least one distributed object is included in the said post-transfer information management table.

14. A distributed processing system according to claim 11, wherein each of said first and second messages further contains an evaluation flag which indicates whether or not a distributed object is under evaluation.

15. A distributed processing system according to claim 14, wherein said distributed object execution control unit executes said distributed object for which said evaluation flag is contained in said first message, in a separate process, when the evaluation flag indicates that said program component for which said evaluation flag is contained in said first message is under evaluation.

16. A distributed processing system according to claim 11, wherein said post-transfer information management table indicates for each of said at least one distributed object an evaluation count which indicates the number of evaluations which are currently performed on

July 22
said each of said at least one distributed object.

17. A distributed processing system according to claim 16, wherein said distributed object execution control unit terminates execution of one of said at least one distributed object in a separate process when said evaluation count corresponding to said one of said at least one distributed object is zero.

10 18. A distributed processing system according to claim 11, further comprising a management server which sets and manages said one or more distributed objects stored in said distributed-object storing repository and said information on said one or more distributed objects
15 held in said pre-transfer information management table.

19. A client apparatus comprising:

an interface unit which controls operations of inputting and outputting software; and

20 a client stub processing unit which executes stub processing of a message which contains version information indicating a program version.

July 22
20. A server apparatus comprising:

25 a server skeleton processing unit which executes skeleton processing of a first message containing version information indicating a program version;

Sub 1
a distributed-object storing repository
which stores one or more distributed objects;

a pre-transfer information management
table which holds information on said one or more
5 distributed objects stored in said distributed-object
storing repository;

a memory which is allocated to an
activated process, and temporarily stores at least one
distributed object transferred from said distributed-
10 object storing repository;

a post-transfer information management
table which holds information on said at least one
distributed object stored in said memory;

a distributed object execution control
15 unit which dynamically links one of said one or more
distributed objects corresponding to said version
information contained in said first message of which
skeleton processing is executed by said server skeleton
processing unit, to said memory, so as to enable execution
20 of at least one function in said one of said one or more
distributed objects; and

a server stub processing unit which
executes stub processing of a second message containing
said version information so as to transmit the second
25 message to another server apparatus.

21. A method for updating a program component

July 12
loaded in a server in an N-tier client-server environment,
comprising the steps of:

(a) storing in said server one or more
program components;

5 (b) holding information on said one or
more program components stored in said server, in a pre-
transfer information management table;

10 (c) transmitting a first message
containing version information indicating a program
version, from a client to said server;

(d) receiving said first message by said
server;

15 (e) dynamically linking one of said one
or more program components corresponding to said version
information contained in said first message, to a memory
which is allocated to an activated process in said server,
so as to enable execution of said one of said one or more
program components in said process;

20 (f) holding in a post-transfer
information management table information on at least one
program component stored in said memory; and

(g) transmitting to another server a
second message containing said version information.

25 22. A method according to claim 21, wherein said
post-transfer information management table indicates for
each of said at least one program component a reference

Sub 5a2
count which indicates the number of executions in which said each of said at least one program component are currently referred to.

5 23. A method according to claim 22, further comprising the step of,

 (h) removing one of said at least one program component from said memory when said reference count for said one of said at least one program component
10 is zero, and information on a newer version of said one of said at least one program component is included in the said post-transfer information management table.

24. A method according to claim 21, wherein each of
15 said first and second messages further contains an evaluation flag which indicates whether or not a program component is under evaluation.

25. A method according to claim 24, wherein, in
20 said step (f), said program component for which said evaluation flag is contained in said first message is executed in a separate process when the evaluation flag indicates that said program component for which said evaluation flag is contained in said first message is
25 under evaluation.

26. A method according to claim 21, wherein said

Sub
Case
post-transfer information management table indicates for each of said at least one program component an evaluation count which indicates the number of evaluations which are currently performed on said each of said at least one
5 program component.

27. A method according to claim 26, further comprising the step of,

10 (i) terminating execution of one of said at least one program component in a separate process when said evaluation count corresponding to said one of said at least one program component is zero.

28. A method according to claim 21, further
15 comprising the step of,

20 (j) setting and managing said one or more program components stored in said server and said information on said one or more program components held in said pre-transfer information management table, by using a management apparatus.

29. A product for use with a program execution apparatus,

25 said product, when used with said program execution apparatus, is able to output control information which directs the program execution apparatus to comprise:
a message receiving unit which receives

a message containing version information indicating a program version;

a program storing unit which stores one or more program components;

5 a pre-transfer information management table which holds information on said one or more program components stored in said program storing unit;

10 a program memory unit which is allocated to an activated process, and temporarily stores at least one program component transferred from said program storing unit;

a post-transfer information management table which holds information on said at least one program component stored in said program memory unit; and

15 a program executing unit which dynamically links one of said one or more program components corresponding to said version information contained in said message received by said message receiving unit, to said program memory unit, so as to
20 enable execution of said one of said one or more program components.

30. A product for use with a server apparatus,

25 said product, when used with said server apparatus, is able to output control information which directs the server apparatus to comprise:

a server skeleton processing unit which

executes skeleton processing of a first message containing version information indicating a program version;

a distributed-object storing repository which stores one or more distributed objects;

5 a pre-transfer information management table which holds information on said one or more distributed objects stored in said distributed-object storing repository;

10 a memory which is allocated to an activated process, and temporarily stores at least one distributed object transferred from said distributed-object storing repository;

15 a post-transfer information management table which holds information on said at least one distributed object stored in said memory;

a distributed object execution control unit which dynamically links one of said one or more distributed objects corresponding to said version information contained in said first message of which skeleton processing is executed by said server skeleton processing unit, to said memory, so as to enable execution of at least one function in said one of said one or more distributed objects; and

20

a server stub processing unit which stub processing of a second message containing said version information so as to transmit the second message to another server apparatus.

25

31. A product for use with a server in an N-tier client-server environment for updating a program component loaded in the server,

said product, when used with said server, is able to
5 output control information which directs the server to execute a method comprising the steps of:

(a) storing in said server one or more program components;

(b) holding information on said one or
10 more program components stored in said server, in a pre-transfer information management table;

(c) receiving a first message containing version information indicating a program version;

(d) dynamically linking one of said one
15 or more program components corresponding to said version information contained in said first message, to a memory which is allocated to an activated process in said server, so as to enable execution of said one of said one or more program components in said process;

(e) holding in a post-transfer
20 information management table information on at least one program component stored in said memory; and

(f) transmitting to another server a second message containing said version information.